WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Software concepts for coupling reduced and detailed simulations

Workshop on Reduced Basis Methods, Ulm

wissen.leben WWU Münster

Martin Drohmann, Bernard Haasdonk,

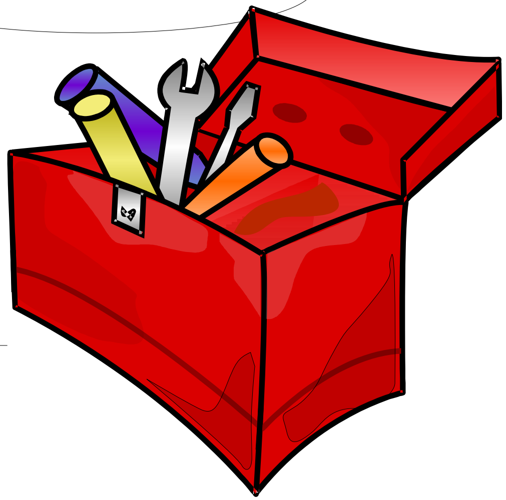
Sven Kaulmann, Mario Ohlberger

Institut für Numerische und Angewandte Mathematik

> Motivation

PDE Discretizations (FEM, FV, DG):
Navier-Stokes, Groundwater-Flow, Convection-Diffusion,
Poisson, Maxwell, ...

Parametrization



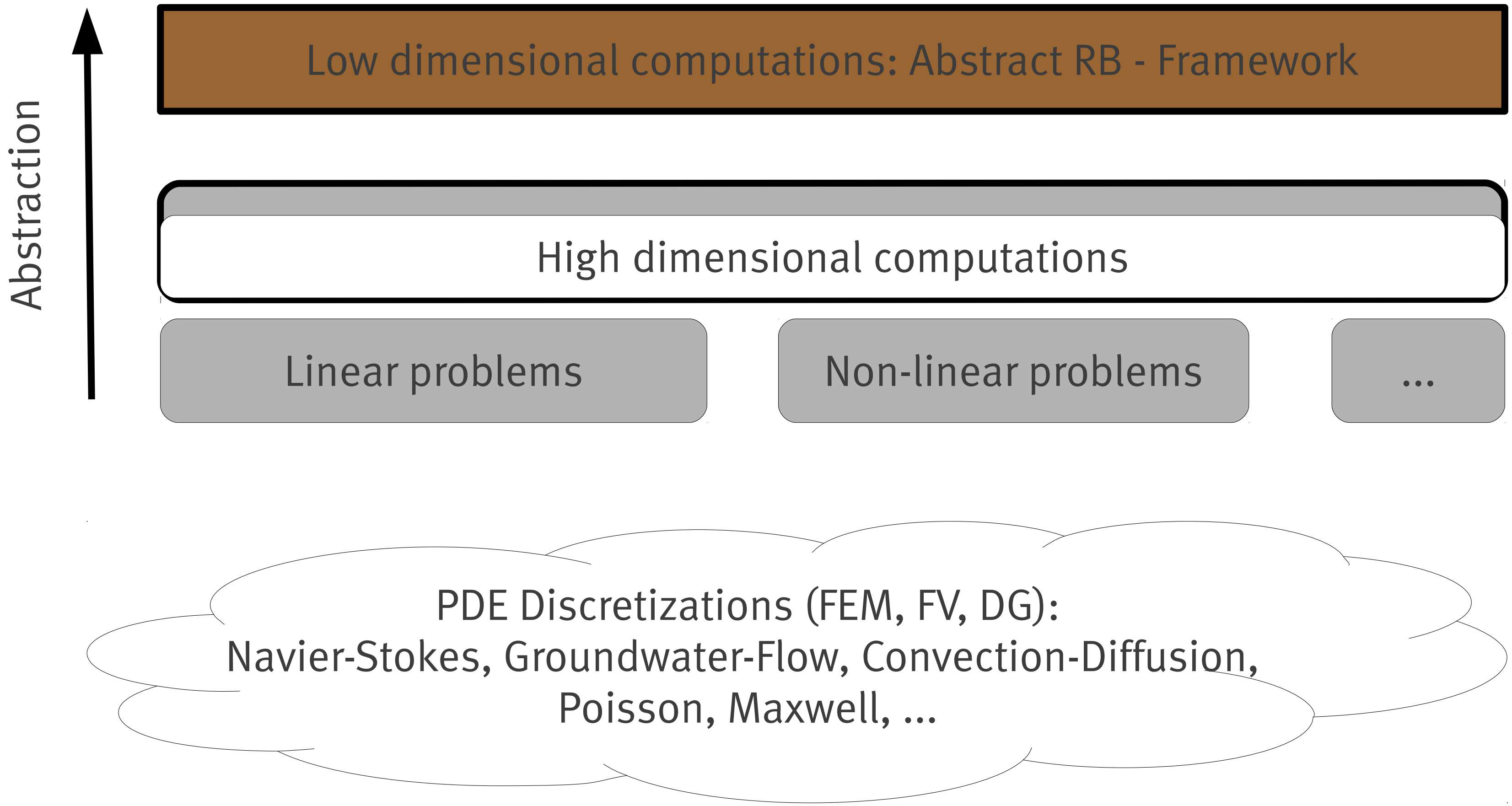
Magical RB-Tools

Reduced model

› Goals Specification

- Abstract Reduced Basis Framework
- Re-use of (existing) efficient implementations of PDE discretizations
- Short time for implementing the detailed simulations as reduced simulations

› Abstract Software Concept



› Low dimensional Computations (RBMATLAB)

- Control of Offline-Algorithms
  - POD-Greedy Algorithm, Empirical Interpolation Algorithm
  - Gathering and post-processing of reduced matrices
- Low-Dimensional Computations
  - Reduced Simulations
  - A posteriori error estimation
- Low dimensional data visualization
- (Implemented in MATLAB)

High dimensional computations (DUNE-RB)

- Storage / Manipulation of reduced spaces
- Efficient high dimensional linear algebra algorithms
  - POD, Orthonormalization, Gram-Matrix computations
- Parametrization
- (Implemented in C++, based on
  - DUNE core modules (http://dune-project.org) and
  - DUNE-FEM (dune.mathematik.uni-freiburg.de)

The Glue

- Communication between DUNE-RB and RBMATLAB can be realized by

    1. compiling DUNE-RB as (mex-) library for matlab or:
    2. TCP/IP Communication between two stand-alone applications

Proof of Concept (linear evolution problem)

```
RBMATLAB > matlab

                  < M A T L A B (R) >
        Copyright 1984-2008 The MathWorks, Inc.
              Version 7.6.0.324 (R2008a)
                   February 10, 2008


  To get started, type one of these: helpwin, helpdesk, or demo
.
  For product information, visit www.mathworks.com.

starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!
>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
```

Proof of Concept (linear evolution problem)

```
                    < M A T L A B (R) >
          Copyright 1984-2008 The MathWorks, Inc.
                Version 7.6.0.324 (R2008a)
                     February 10, 2008


  To get started, type one of these: helpwin, helpdesk, or demo
.
  For product information, visit www.mathworks.com.

starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!
>>
>> [a,b,c] = ...
mexclient('echo', [1, 2], ....
                  struct('field', [1,2]), ...
                  { [1,2], [3,4] });
client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1
copying argument no. 0
copying argument no. 1
copying argument no. 2
>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'echo'
with 4 arguments and 3 return values.
=======================================================

copying argument no. 0
copying argument no. 1
copying argument no. 2
```

Proof of Concept (linear evolution problem)

```
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!
>>
>> [a,b,c] = ...
mexclient('echo', [1, 2], ....
                  struct('field', [1,2]), ...
                  { [1,2], [3,4] });
client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1
copying argument no. 0
copying argument no. 1
copying argument no. 2
>> a

a =

     1     2

>> b

b =

    field: [1 2]

>> c

c =

    [1x2 double]    [1x2 double]

>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'echo'
with 4 arguments and 3 return values.
======================================================

copying argument no. 0
copying argument no. 1
copying argument no. 2
```

Proof of Concept (linear evolution problem)

```
                    < M A T L A B (R) >
          Copyright 1984-2008 The MathWorks, Inc.
                Version 7.6.0.324 (R2008a)
                     February 10, 2008


  To get started, type one of these: helpwin, helpdesk, or demo
.
  For product information, visit www.mathworks.com.

starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!
>>
>>
>> % load model parameters
>>
>> model = convdiff_dune_model;
Warning: Name is nonexistent or not a directory: mexclient.
> In path at 110
  In addpath at 87
  In convdiff_dune_model at 95
client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1
>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'init_model'
with 1 arguments and 1 return values.
=======================================================

read discfunclist_xdr from headerfile, size = 20
Using the explicit ode solver! In order to use a different disc
retization, change the 'DISCRETIZATION' make variable

Received call for processing 'get_mu'
with 1 arguments and 1 return values.
=======================================================


Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
=======================================================
```

Westfälische Wilhelms-Universität Münster

Proof of Concept (linear evolution problem)

```
> In path at 110
  In addpath at 87
  In convdiff_dune_model at 95
client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1
>>
>> model.rb_problem_type

ans =

lin_evol
>> model.RB_generation_mode

ans =

greedy_uniform_fixed
>> model.RB_stop_Nmax

ans =

    20
>> model.T                        % this is read from DUNE-RB <<<<<<

ans =

     1

>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'init_model'
with 1 arguments and 1 return values.
=======================================================

read discfunclist_xdr from headerfile, size = 20
Using the explicit ode solver! In order to use a different disc
retization, change the 'DISCRETIZATION' make variable

Received call for processing 'get_mu'
with 1 arguments and 1 return values.
=======================================================


Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
=======================================================
```

Proof of Concept (linear evolution problem)

```
ans =

lin_evol

>> model.RB_generation_mode

ans =

greedy_uniform_fixed

>> model.RB_stop_Nmax

ans =

    20

>> model.T                   % this is read from DUNE-RB <<<<<<

ans =

     1

>>
>>
>>
>>
>> % generate high dimensional model specific data, like e.g.
>> % the grid
>>
>> model_data = gen_model_data(model);
>>
>>
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defa
ulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'init_model'
with 1 arguments and 1 return values.
=======================================================

read discfunclist_xdr from headerfile, size = 20
Using the explicit ode solver! In order to use a different disc
retization, change the 'DISCRETIZATION' make variable

Received call for processing 'get_mu'
with 1 arguments and 1 return values.
=======================================================


Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
=======================================================


Received call for processing 'gen_model_data'
with 1 arguments and 1 return values.
=======================================================
```

Proof of Concept (linear evolution problem)

```
>> % Just for fun: Do a DETAILED simulation in DUNE-RB >>>>>>
>>
>> %   first set the parameter mu
>> model = set_mu(model, [0.0 0.5 1.0]);
>>
>> %   then run the simulation
>> tic; sim_data = detailed_simulation(model, model_data); toc
Elapsed time is 11.795319 seconds.
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
```

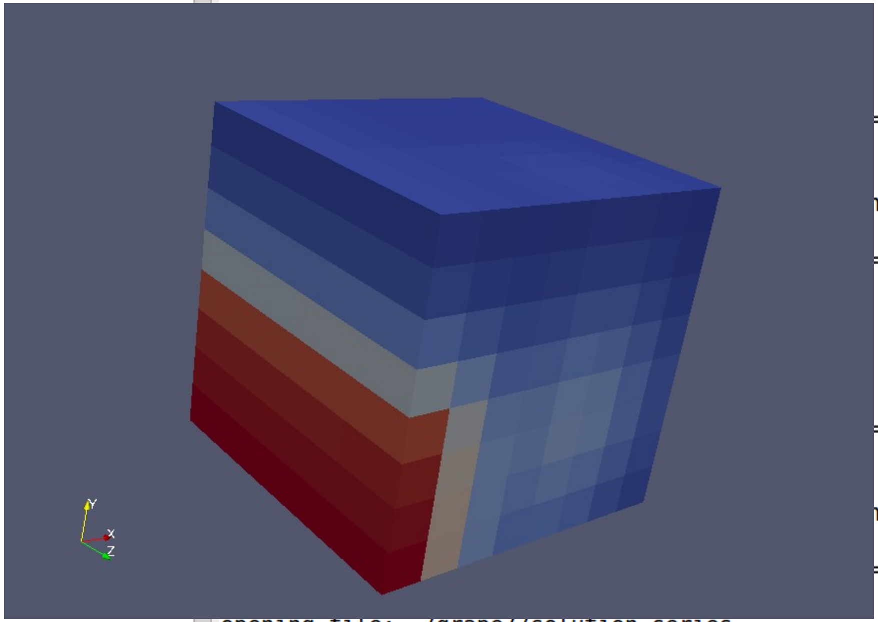
```
Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
=============================================================


Received call for processing 'gen_model_data'
with 1 arguments and 1 return values.
=============================================================


Received call for processing 'set_mu'
with 2 arguments and 0 return values.
=============================================================


Received call for processing 'detailed_simulation'
with 1 arguments and 1 return values.
=============================================================

opening file: ./grape//solution.series

Received call for processing 'set_mu'
with 2 arguments and 0 return values.
=============================================================


Received call for processing 'detailed_simulation'
with 1 arguments and 1 return values.
=============================================================

opening file: ./grape//solution.series
```

Proof of Concept (linear evolution problem)

```
>> % Just for fun: Do a DETAILED simulation in DUNE-RB >>>>>>
>>
>> %   first set the parameter mu
>> model = set_mu(model, [0.0 0.5 1.0]);
>>
>> %   then run the simulation
>> tic; sim_data = detailed_simulation(model, model_data); toc
Elapsed time is 11.795319 seconds.
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
```

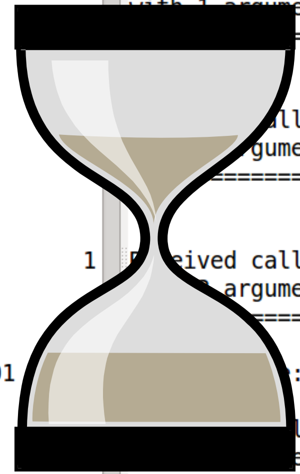
11.8 seconds!

```
Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
===========================================================

Received call for processing 'gen_model_data'
with 1 arguments and 1 return values.
===========================================================
opening file: ./grape//solution.series
```



Proof of Concept (linear evolution problem)

```
>> % Generate the reduced basis with the POD-Greedy algorithm
>> % in DUNE-RB >>>>>>>>
>>
>> detailed_data = gen_detailed_data(model, model_data);
Starting RB extension loop

Detected maximum error prediction 0.044006 for mu=[0.001
    1         0.5]
Extended RB to length 2

Detected maximum error prediction 0.015456 for mu=[0  1
Extended RB to length 3

Detected maximum error prediction 0.012877 for mu=[0
1         0.5]
Extended RB to length 4

Detected maximum error prediction 0.01064 for mu=[0          1
         0.5]
Extended RB to length 5

Detected maximum error prediction 0.0084073 for mu=[0.001
   0.5          1]
Extended RB to length 6

Detected maximum error prediction 0.0073233 for mu=[0  1  1]
Extended RB to length 7

Detected maximum error prediction 0.0055443 for mu=[0  1  1]
Extended RB to length 8

Detected maximum error prediction 0.0048443 for mu=[0
 1         0.5]
```



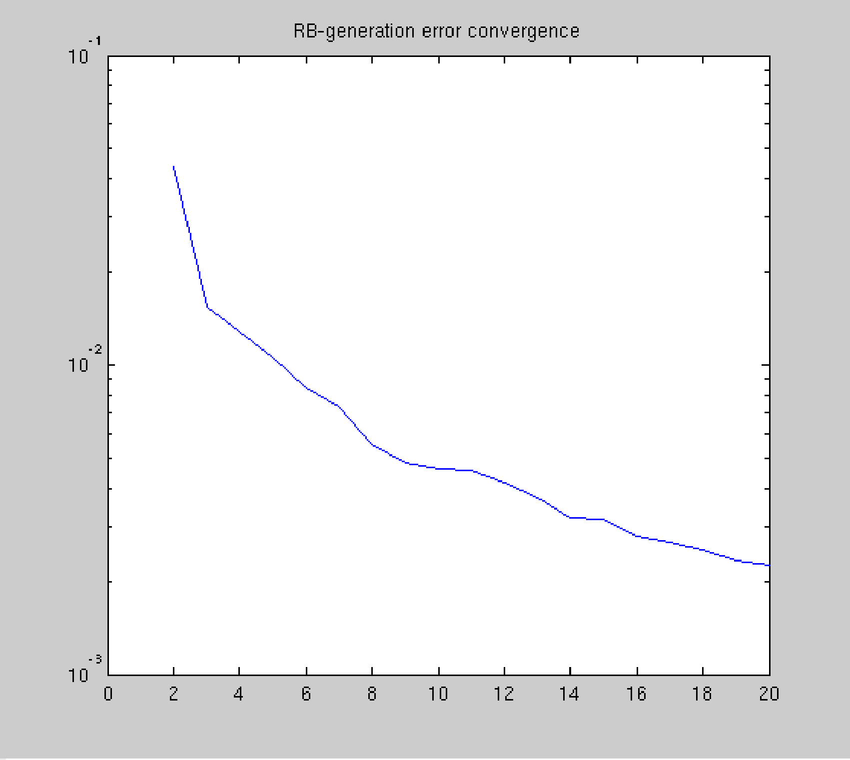
```
Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
======================================================

Received call for processing 'get_mu'
with 1 arguments and 1 return values.
======================================================

    ll for processing 'set_mu'
    guments and 0 return values.
======================================================

  eived call for processing 'rb_extension_PCA'
   arguments and 1 return values.
======================================================

       : ./grape//solution.series

      l for processing 'set_mu'
      ents and 0 return values.
======================================================

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
======================================================
```

Westfälische
Wilhelms-Universität
Münster

Proof of Concept (linear evolution problem)

```
>>
>>
>> detailed_data.RB_info

ans =

                      M_train: [3x64 double]
             max_err_sequence: [1x20 double]
                  mu_sequence: [3x20 double]
                   mu_ind_seq: [1x20 double]
           toc_value_sequence: [1x20 double]
                 M_first_errs: [64x1 double]
           stopped_on_epsilon: 0
    stopped_on_max_val_train_ratio: 0
           stopped_on_timeout: 0
              stopped_on_Nmax: 1
       stopped_on_empty_extension: 0
     stopped_on_Nlimit_estimation: 0
                  M_last_errs: [64x1 double]
                 elapsed_time: 3.0000e-06

>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
```

```
Received call for processing 'is_valid_rb'
```




```
opening file: ./grape//solution.series
```

wissen.leben
WWU Münster

Proof of Concept (linear evolution problem)

```
>> % NOW: Generate reduced matrices for online computations
>> % and get them to RBMATLAB <<<<<<<<
>>
>> reduced_data = gen_reduced_data(model, detailed_data)

reduced_data =

      a0: {[1x20 double]}
    LL_I: {2x1 cell}
    LL_E: {5x1 cell}
      bb: {4x1 cell}
    K_II: {4x1 cell}
    K_IE: {10x1 cell}
    K_EE: {25x1 cell}
     m_I: {8x1 cell}
     m_E: {20x1 cell}
       m: {16x1 cell}
       N: 20

>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
```

```
Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
======================================================


Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
======================================================


Received call for processing 'set_mu'
with 2 arguments and 0 return values.
======================================================


Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.
======================================================

opening file: ./grape//solution.series

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
======================================================


Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
======================================================
```

Proof of Concept (linear evolution problem)

```
>> % The matrices are all of small sizes: e.g. the explicit
>> % discretization operator: LL_E
>>
>> sizes=cellfun(@(X) size(X), reduced_data.LL_E, ...
                               'UniformOutput', false);
>> sizes{:}

ans =

    20    20


ans =

    20    20


ans =

    20    20


ans =

    20    20


ans =

    20    20

>>
>>
```

```
Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
=====================================================


Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
=====================================================


Received call for processing 'set_mu'
with 2 arguments and 0 return values.
=====================================================


Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.
=====================================================

opening file: ./grape//solution.series

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
=====================================================


Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
=====================================================
```

Proof of Concept (linear evolution problem)

```
>> % Now fast reduced simulations are possible in RBMATLAB
>> % without any communication to DUNE-RB
>>
>> model = model.set_mu(model, [0 0.5 1], true);
>>
>> tic; rb_sim_data=rb_simulation(model, reduced_data); toc
Elapsed time is 0.020223 seconds.
>>
>>
>> rb_sim_data

rb_sim_data =

        a: [20x113 double]
    Delta: [1x113 double]
     LL_I: [20x20 double]
     LL_E: [20x20 double]

>>
>>
>> % Error estimator at end time:
>>
>> rb_sim_data.Delta(end)

ans =

    0.0019

>>
>>
>>
>>
>>
```

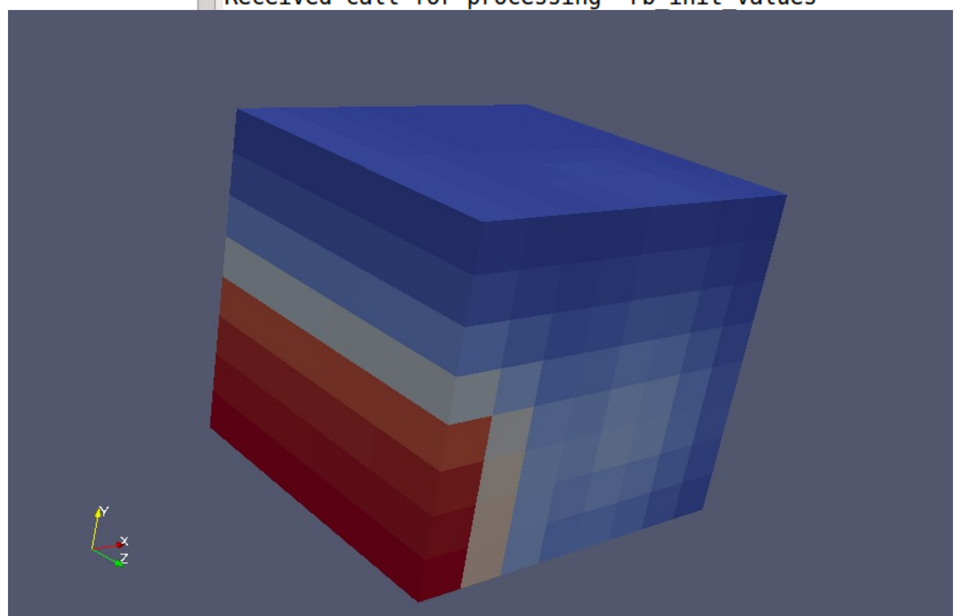
0.02 seconds!

```
Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
======================================================

Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
======================================================

Received call for processing 'set_mu'
with 2 arguments and 0 return values.
======================================================

Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.
======================================================

opening file: ./grape//solution.series

Received call for processing 'rb_init_values
with 2 arguments and 1 return values.
======================================================

Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
======================================================
```

Proof of Concept (linear evolution problem)

```
>> % Of course, the solution can be reconstructed in
>> % DUNE-RB >>>>>>>>>
>>
>> tic;...
dummy=rb_reconstruction(model, detailed_data, rb_sim_data); toc
Elapsed time is 0.359108 seconds.
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
```

```
Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.
=====================================================

opening file: ./grape//solution.series

Received call for processing 'rb_init_values'
```

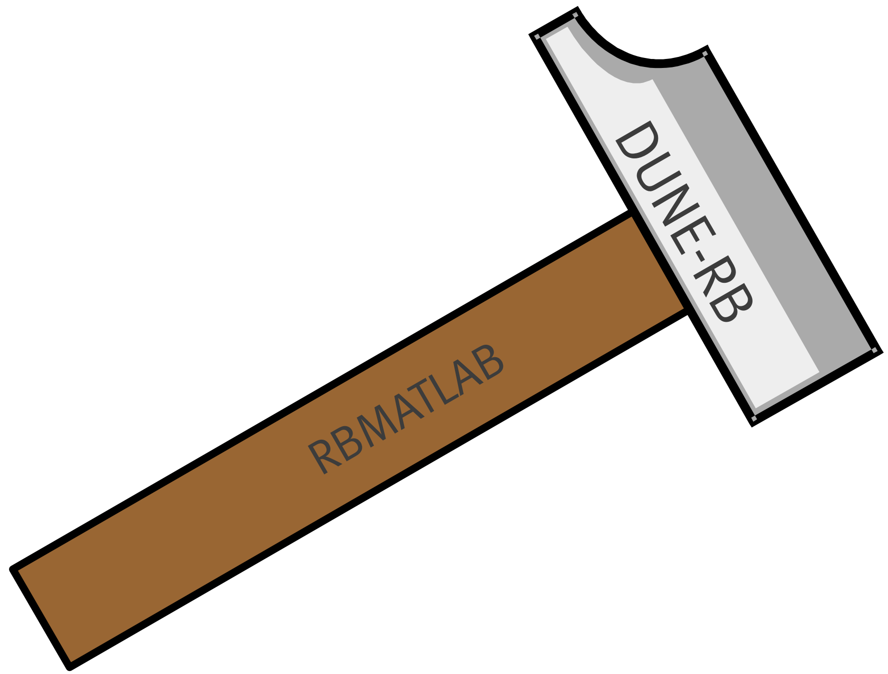


Linear Transport Problem

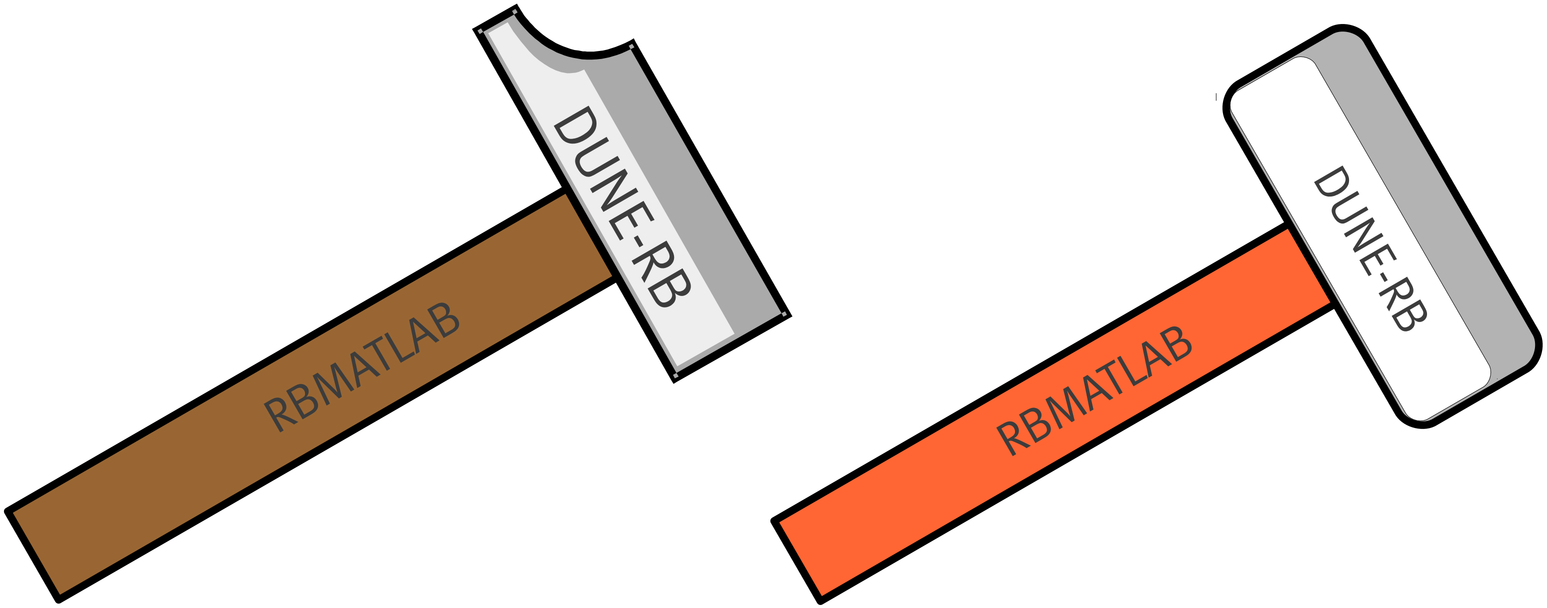
	High-dim. Solution (sec)	RB Generation	Gen. of Online Matrices	Reduced Sim.	Reconstruction	Grid Cells	$L^\infty - L^2$-Error[1]
2D Transport (25 Base Functions)	11	71	6.69	0.11	0.33	1,024	$1.42 \cdot 10^{-3}$
2D Transport (50 Base Functions)	11	2,250	21	0.15	0.42	1,024	$4.64 \cdot 10^{-4}$
3D Transport (50 Base Functions)	944	$1.57 \cdot 10^5$	4,659	0.15	26	32,768	$9.11 \cdot 10^{-4}$

Table 1: Numerical results for a transport problem in 2D and 3D with non-divergence-free velocity.

So, we have a hammer for linear evolution problems ...

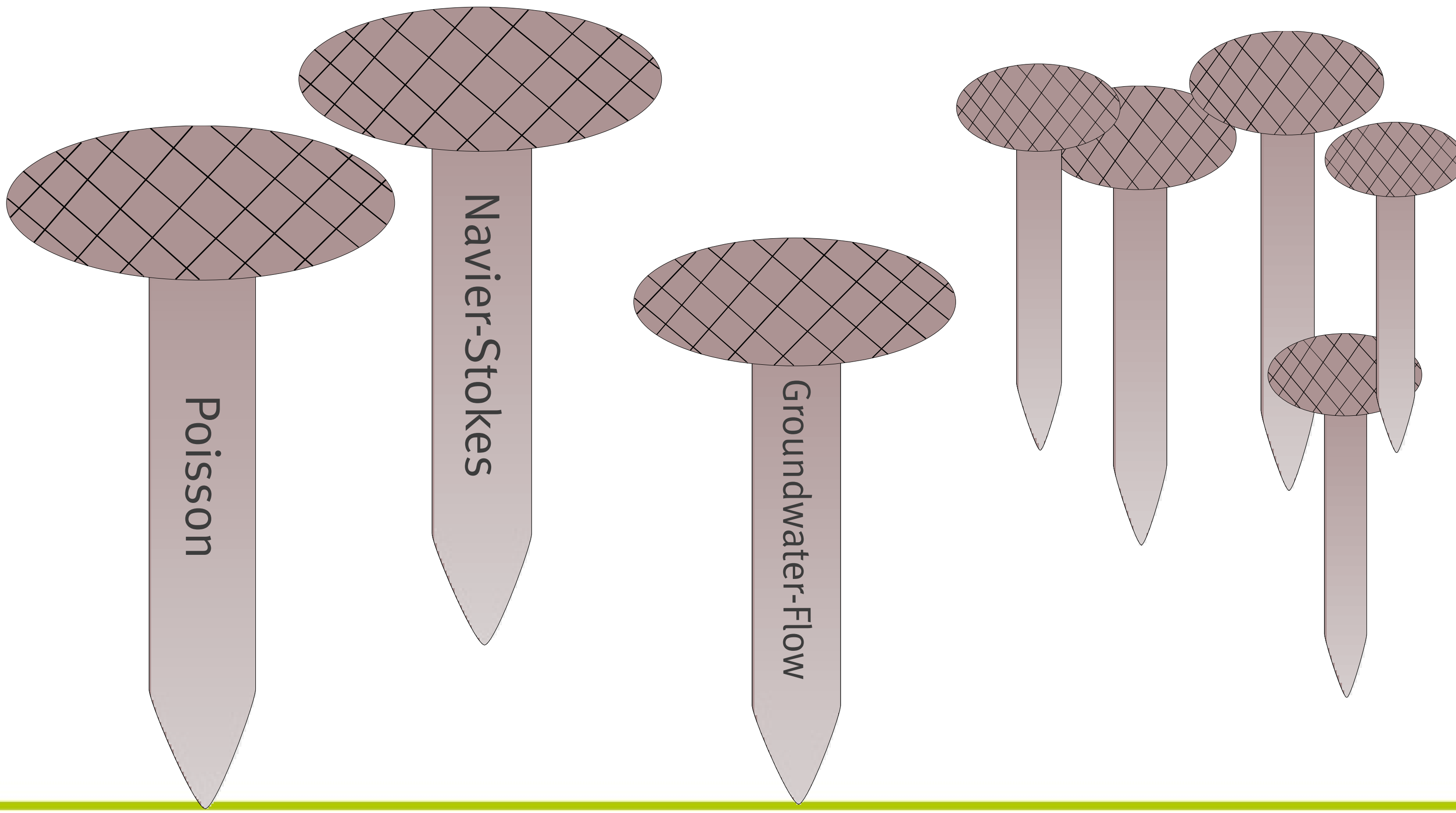


… and even one for non-linear problems …



Westfälische
Wilhelms-Universität
Münster

But how do you make your problems look like a nail?



Interface to PDE descretizations

```
>> % Generate the reduced basis with the POD-Greedy algorithm
>> % in DUNE-RB >>>>>>>>
>>
>> detailed_data = gen_detailed_data(model, model_data);
Starting RB extension loop

Detected maximum error prediction 0.044006 for mu=[0.001
    1         0.5]
Extended RB to length 2

Detected maximum error prediction 0.015456 for mu=[0   1   1]
Extended RB to length 3

Detected maximum error prediction 0.012877 for mu=[0
1         0.5]
Extended RB to length 4

Detected maximum error prediction 0.01064 for mu=[0            1
         0.5]
Extended RB to length 5

Detected maximum error prediction 0.0084073 for mu=[0.001
   0.5          1]
Extended RB to length 6

Detected maximum error prediction 0.0073233 for mu=[0   1   1]
Extended RB to length 7

Detected maximum error prediction 0.0055443 for mu=[0   1   1]
Extended RB to length 8

Detected maximum error prediction 0.0048443 for mu=[0
 1         0.5]
```

```
Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
=====================================================


Received call for processing 'get_mu'
with 1 arguments and 1 return values.
=====================================================


Received call for processing 'set_mu'
with 2 arguments and 0 return values.
=====================================================


Received call for processing 'rb_extension_PCA'
with 3 arguments and 1 return values.
=====================================================

opening file: ./grape//solution.series

Received call for processing 'set_mu'
with 2 arguments and 0 return values.
=====================================================


Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
=====================================================
```

Interface to PDE discretizations (linear)

- For PDE discretization of form

For $\boldsymbol{\mu} \in \mathcal{P}$ compute $\left\{u_h^k(\boldsymbol{\mu})\right\}_{k=0}^{K} \subset \mathcal{W}_h \subset L^2(\Omega)$ by

$$u_h^0(\boldsymbol{\mu}) := P[u_0(\boldsymbol{\mu})]$$

$$u_h^k(\boldsymbol{\mu}) - \Delta t \mathcal{L}^I(\boldsymbol{\mu})[u_h^k(\boldsymbol{\mu})] := u_h^{k-1}(\boldsymbol{\mu}) + \Delta t \mathcal{L}^E(\boldsymbol{\mu})[u_h^{k-1}(\boldsymbol{\mu})].$$

- Implement affinely parameter dependent operators

$$\mathcal{L} = \sum_{q=1}^{Q_L} \boxed{\sigma_L^q(\boldsymbol{\mu})} \, \boxed{\mathcal{L}^q}$$

Interface to PDE discretizations (linear)

Affinely parametrized operators

$$\mathcal{L} = \sum_{q=1}^{Q_L} \boxed{\sigma_L^q(\mu)}\, \boxed{\mathcal{L}^q}$$

with methods

- Parametrization [`set_mu()`]
- Parameter dependent coefficients [`coefficient()`]
- Parameter independent operators [`component()`]

Interface to PDE discretizations (linear)

Affinely parametrized operators

$$\mathcal{L} = \sum_{q=1}^{Q_L} \boxed{\sigma_L^q(\mu)}\, \boxed{\mathcal{L}^q}$$

with methods

- Parametrization [`set_mu()`]
- Parameter dependent coefficients [`coefficient()`]
- Parameter independent operators [`component()`]

⇨ Automatic generation of reduced matrices:

$$(\mathbf{L}^q)_{n,m} = \int_\Omega \mathcal{L}^q [\varphi_m]\, \varphi_n, \qquad q = 1, \ldots, Q_L, m, n = 1, \ldots, N$$

WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

Interface to PDE discretizations (non-linear)

- For PDE discretizations of the form

For $\boldsymbol{\mu} \in \mathcal{P}$ find $\{u_h\}_{k=0}^K \subset \mathcal{W}_h \subset L^2(\Omega)$, s.t.

$$u_h^0 := P_h\left[u_0(\boldsymbol{\mu})\right], \qquad u_h^{k+1} := u_h^{k+1,\nu_{\max}(k)}$$

with Newton iteration

$$u_h^{k+1,0} := u_h^k, \qquad u_h^{k+1,\nu+1} := u_h^{k+1,\nu} + \delta_h^{k+1,\nu+1},$$

$$\left(\mathrm{Id} + \Delta t \mathbf{D}\mathcal{L}_h^I\big|_{u_h^{k+1,\nu}}\right)\left[\delta_h^{k+1,\nu+1}\right] = u_h^k - u_h^{k+1,\nu} - \Delta t\left(\mathcal{L}_h^I\left[u_h^{k+1,\nu}\right] + \mathcal{L}_h^E\left[u_h^k\right]\right).$$

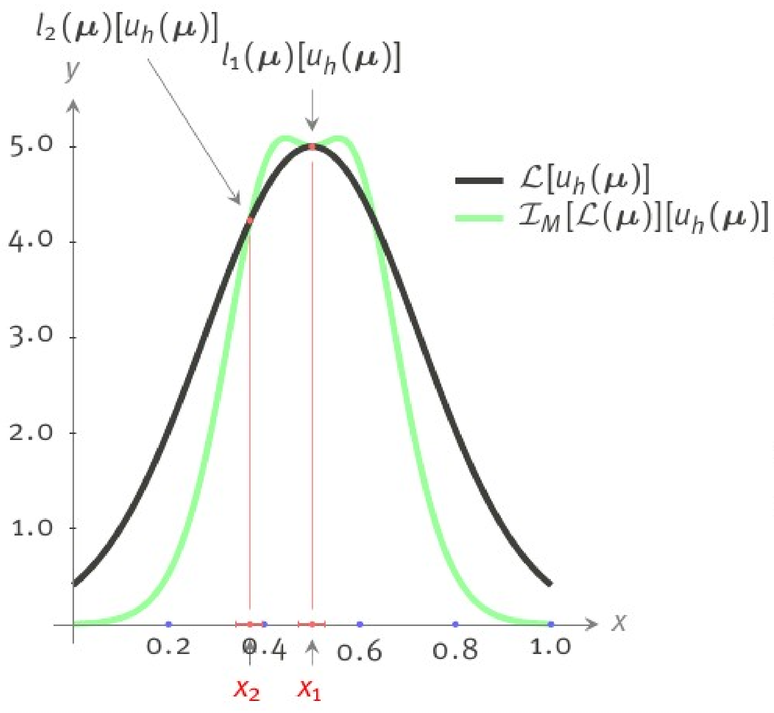
- Implement empirical interpolation for discrete operators

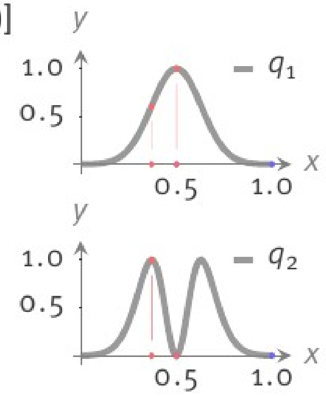
$$\mathcal{L}_h(\boldsymbol{\mu})\left[u_h^k(\boldsymbol{\mu})\right] \approx \mathcal{I}_M\left[\mathcal{L}_h(\boldsymbol{\mu})\right]\left[u_h^k(\boldsymbol{\mu})\right] = \sum_{m=1}^M l_m(\boldsymbol{\mu})\left[u_h^k(\boldsymbol{\mu})\right] \xi_m$$

wissen.leben WWU Münster

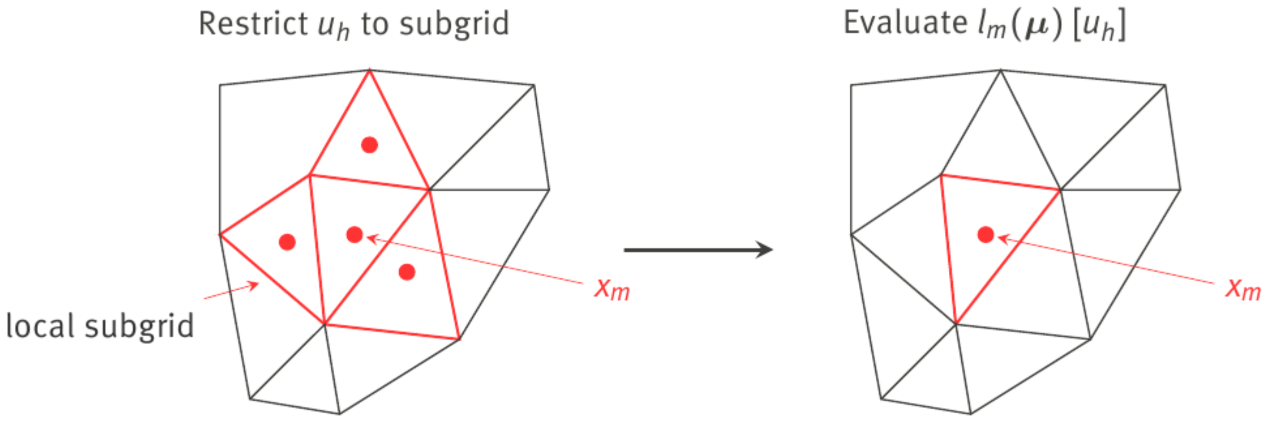
The coefficient functionals $l_m(\boldsymbol{\mu}) = \mathcal{L}_h(\boldsymbol{\mu})[\cdot](x_m)$ can be computed efficiently, if

- Operator has localized structure (small stencil) and
- Local geometry information can be precomputed in offline phase.





Empirical Interpolation for Operators (local grid)



Automatic generation of localized operators

Usual dependencies for local operator evaluations:

$$l_m(\mu) = \mathcal{L}_h(\mu)[\cdot](x_m)$$

Basis function evaluations at quadrature points

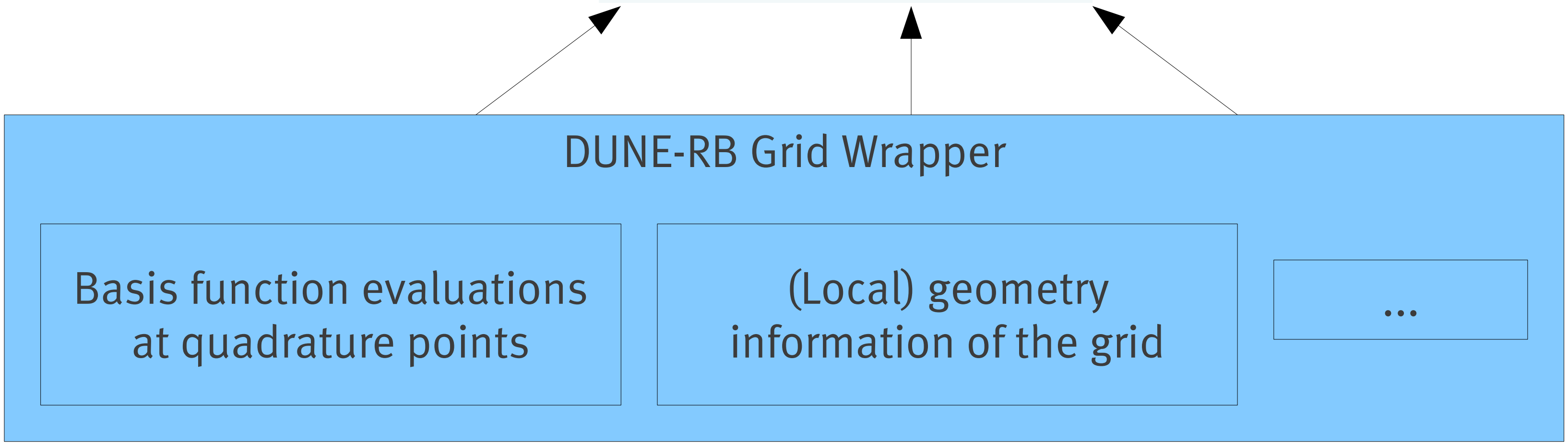
(Local) geometry information of the grid

...

Automatic generation of localized operators

Therefore: Delegation of function calls concerning grid geometry and discrete function space to a wrapper:

$$l_m(\boldsymbol{\mu}) = \mathcal{L}_h(\boldsymbol{\mu})[\cdot](x_m)$$



Behaviour of DUNE-RB Grid Wrapper

- During detailed simulation:

  Delegate calls directly to the grid and the discrete function space

- During offline phase:

  store all grid and discrete function space information concerning the subgrid in low-dimensional data structures

- During online phase:

  Delegate calls to low-dimensional data structures

Conclusion

- Generic implementation of reduced base tools for linear problems is possible
- Empirical interpolation for non-linear problems can be realized with subgrid extraction
- Future work:
  - More examples
  - Tarball – Release with installation instructions is planned, and will be published on http://morepas.org.

Thank you for your attention!